

design ideas

Edited by Bill Travis and Anne Watson Swager

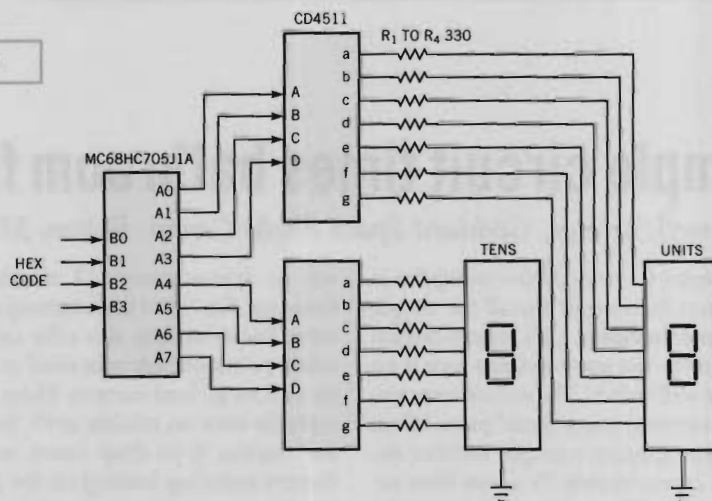
μC visualizes hex code

Abel Raynus, Armatron International, Melrose, MA

IN μC SYSTEMS, information exchanges usually use the hexadecimal 1-2-4-8 format; output data also appear in this format. Reading the hex code is not a problem; several LEDs connected to the output lines can display the answer. The problem arises when you wish to observe the output data. Many engineers are unfamiliar with hex code and prefer to observe data in the common decimal format. If the value of the output data is less than 10, you can use an ordinary BCD-to-seven-segment decoder to visualize the hex code on an LCD or an LED display. But what do you do if the value of the output data is greater than 10? Unfortunately, no decoders can transform hex code into two seven-segment codes. Of course, you could configure such a decoder using a number of logic gates, but another simple and inexpensive option is available. The key to this option is using a low-end μC to transform the hex code into two BCD codes. In Figure 1, the data displayed ranges from 0 to 15. Thus, you need only 4-bit hex code, using four input and eight output μC lines.

Figure 1 uses the approximately \$1 Motorola MC68HC705J1A, with 14 I/O pins. The input lines Pin B0 to Pin B3 re-

Figure 1



You can visualize hex code in BCD format by using this simple circuit.

LISTING 1-HEX-TO-BCD CONVERSION

```

0000      1  * HEX CODE TO 2 BCD CODE TRANSFER*
0000      2  $pagewidth 160
MC68HC705J1A 3  $include "std-j1a.asm" ;standard frame for

4  *VARIABLES
00C0      5      org      RAM
00C0      6  M1      rmb      1
7  *INITIALIZATION
07F1      8      org      MOR;
07F1      9      fcb      %00100000;resistor osc
0300     10      org      ROM
0300 [02] A600    11  init   lda      #$00 ; prtB as input
0302 [04] B705    12      sta      ddrB
0304 [02] A6FF    13      lda      #$ff ; prtA as output
0306 [04] B704    14      sta      ddrA
0308 [05] 3F00    15      clr      prtA ; 0 -> prtA
16 *****
030A [03] B601    17  main   lda      prtB
030C [02] A40F    18      and      #%00001111 ; extract data from pB0-pB3
030E [04] B7C0    19      sta      M1
0310 [02] A109    20      cmp      #9          ; data > 9?
0312 [03] 220A    21      bhi      h1
0314 [02] A600    22      lda      #$00          ;set 00 to pA4-pA7 (tens)
0316 [04] B700    23      sta      prtA
0318 [03] BAC0    24      ora      M1
031A [04] B700    25      sta      prtA          ;set data to pA0-pA3
(units)
031C [03] 20EC    26      bra      main
031E [02] A00A    27  h1     sub      #9a          ; (data-10) -> Acc
0320 [04] B7C0    28      sta      M1
0322 [02] A610    29      lda      #$10          ;set 1 to pA4-pA7 (tens)
0324 [03] 20F0    30      bra      h2
31 *****
07FE      32      org      VECTORS+6
07FE      33      fdb      init
    
```

μC visualizes hex code.....	133
Simple circuit times bathroom fan.....	134
Termination supply tracks one-half core voltage.....	136
Switch debouncer isolates input and output.....	138
Circuit detects reset source.....	140
Interface LCD with ease	142

ceive the 4-bit hex code to be displayed. The μ C-assembly program in **Listing 1**, converts the hex code into two BCD codes, which appear on lines Pin A0 to Pin A3 (units) and Pin A4 to Pin A7 (tens). These outputs drive the two standard BCD-to-seven-segment decoders, which, in turn, drive the common-cathode LED displays. You can use the same

method for an expanded data range, but you need more I/O lines, decoders, and displays. For example, the 8-bit hex code covers the data range 0 to 255, but it needs eight input lines, 12 output lines, and three decoders and displays. You can download **Listing 1** and the "include" file in line 3 from EDN's Web site, www.ednmag.com. Click on "Search Databases"

and then enter the Software Center to download the file for Design Idea #2518. (DI #2518)

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 301

Simple circuit times bathroom fan

Maxwell Strange, Goddard Space Flight Center, Fulton, MD

FORGET TO TURN off the ceiling fan in your bathroom? Install the simple timer in **Figure 1**. It's located out-of-sight in the fan unit, and you turn it on via the wall switch. The circuit costs virtually nothing, using "junk" parts. When ac power appears, a simple rectifier develops approximately 7V across filter capacitor C. This voltage powers the LM2905 analog timer and simultaneously triggers it via trigger-input Pin 1; output Pin 7 then goes low, turning on the solid-state relay and fan. Low-leakage capacitor C_T and resistor R_T set the time delay; for the values shown, the delay is 1000

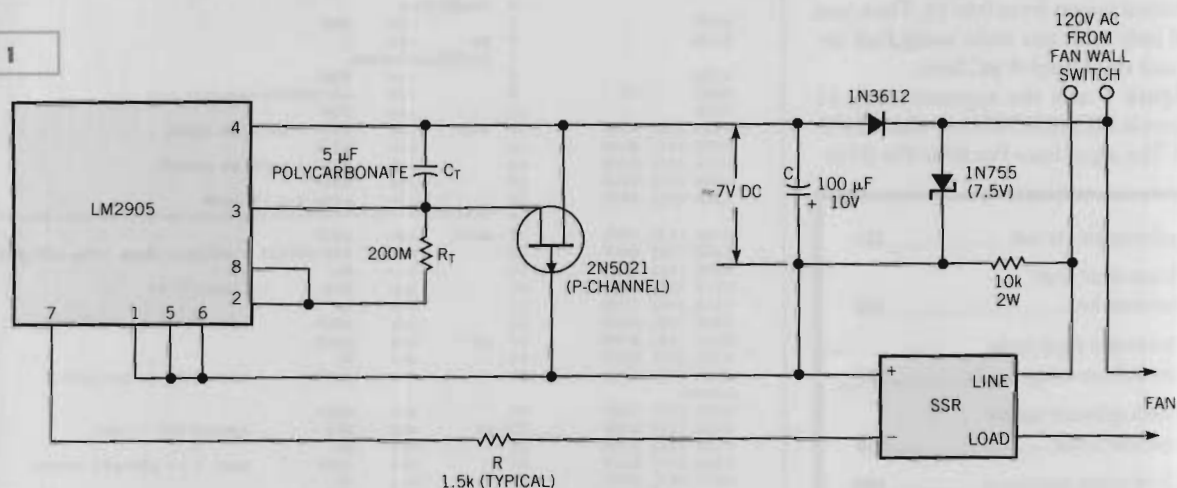
sec, or approximately 17 minutes. At time-out, Pin 7 goes high, turning off the relay. For most fans, this relay can be a small, pc-mountable unit rated at as little as 0.5A ac load current. These relays typically turn on reliably at 3V. You can use resistor R to drop excess voltage, thereby reducing loading on the power supply.

The p-channel JFET turns on when the power switches off, rapidly discharging C_T to allow immediate recycling. The 10-k Ω series resistor makes the circuit inherently safe; in a worst-case failure, line voltage appears across this resistor and

develops a harmless 1.4W. You can easily adapt this circuit to other applications. You can change the timing or make it linearly adjustable, and you can program the solid-state relay to turn on instead of off at time-out, by connecting Pin 8 of the timer to Pin 4 instead of Pin 2. You can also reverse the action of the solid-state relay by reversing the control inputs. (DI #2520)

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 302

Figure 1



Gain control of your bathroom fan by using this simple time-out circuit.

nal error amplifier. R_4 sets the gain of the internal amplifier to unity. R_5 forces a soft start on turn-on and also eliminates overshoot. IC_{1C} amplifies the 1.23V internal reference to approximately 2.5V, which sets the operating point for the error amplifier, IC_{1A} .

The 2.5V also connects to the sense in-

put of the LM2636, which would normally sense the regulated output voltage. Because the termination output voltage must be variable (to track the core voltage), a fixed 2.5V goes to the sense pin, and the control pins 16 and 18 are grounded. These connections program the internal DAC for a 2.5V output. This

scheme prevents the LM2636's internal error-detection circuitry from shutting down the part in response to an under-voltage or overvoltage condition. R_3 , R_6 , C_1 , and C_2 provide loop compensation. (DI #2517)

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 303

Switch debouncer isolates input and output

Phill Leyva, Maxim Integrated Products, Sunnyvale, CA

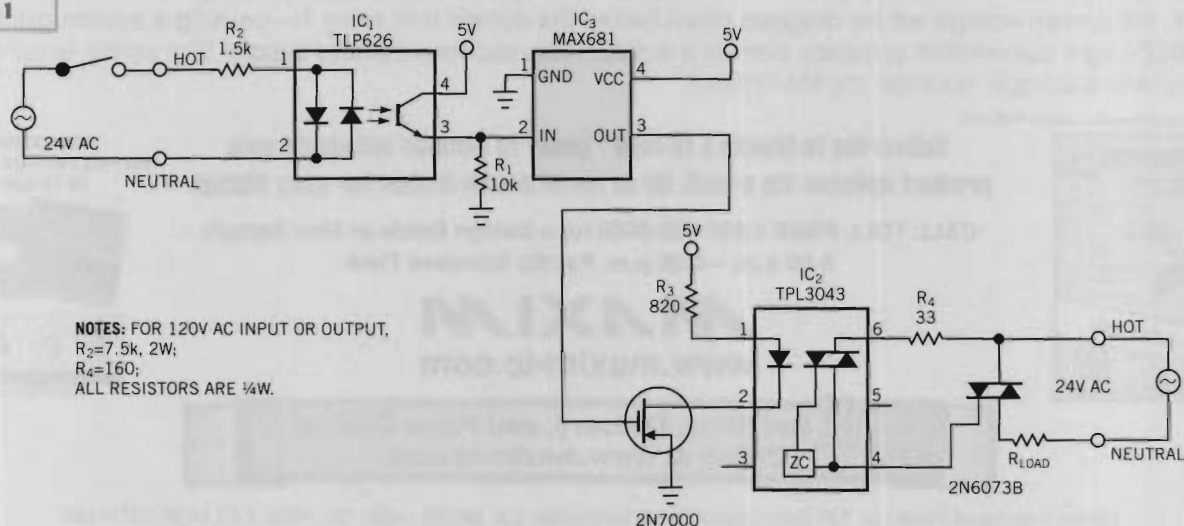
MODERN PROGRAMMABLE-logic controllers (PLCs) for automated process-control systems have either 16 or 32 inputs and accept ac voltages of 24 to 120V. A single circuit (a relay for isolation and an RC network with a Schmitt trigger to debounce signals to the processor) can debounce all the PLC signals in sequence. However, this approach slows real-time data processing. Such debounce circuits also produce delay times that change with relay wear and capacitor aging. In the PLC program, you can use a debounce timer for each input,

but this technique increases the program-scan time and ties up valuable timers. The solid-state, electrically isolated circuit in **Figure 1** debounces single inputs without slowing the PLC module. Optoisolators IC_1 and IC_2 provide electrical isolation for the ac sources at the input and output. IC_3 is a CMOS switch debouncer whose output (a 4V logic high) appears following a fixed 40-msec delay. A 63-k Ω pullup resistor, connected internally between IN and V_{CC} , forms a voltage divider with R_1 . R_1 's value ensures a logic low of less than 0.8V at IN when

IC_1 's optotransistor (an emitter follower) is off.

The two LEDs in IC_1 , which illuminate the phototransistor on alternate half cycles of the ac-input current, rectify this current. Most optoisolator applications set the current-transfer ratio (CTR) to more than 10 to ensure an accurate reproduction of the input signal. The circuit in **Figure 1**, however, sets the CTR to less than 1, which ensures that the emitter follower does not turn off as the ac current goes to zero twice in each cycle. R_2 biases the emitter follower such

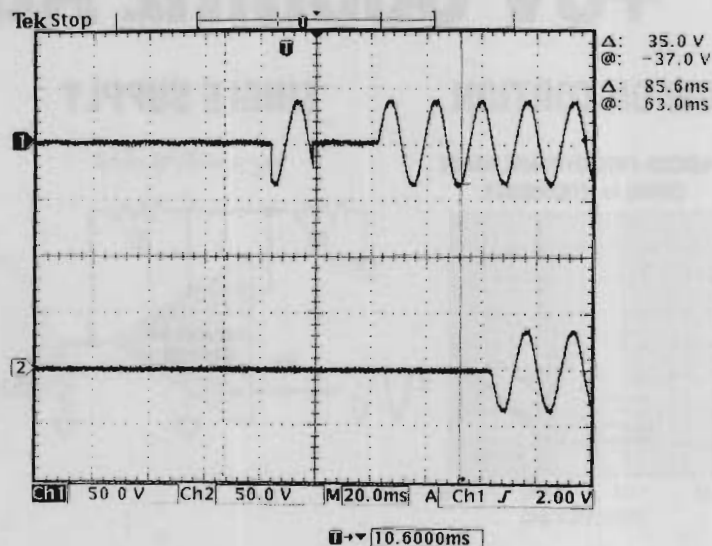
Figure 1



This debouncer circuit allows an isolated ac voltage to control a separately isolated ac source.

that IC₃'s IN signal remains greater than the high level, 2.4V, during these zero crossings. This action eliminates the capacitor normally found in debouncing circuits. IC₃'s OUT pin drives the n-channel MOSFET, and R₃'s value is such that the resulting current flow (in the MOSFET and LED) is approximately 5 mA. When the MOSFET turns on, the LED activates IC₂'s zero-crossing triac driver. Thus, when the power triac turns on, an ac source connected to the output drives R_{LOAD} with as much as 4A (Figure 2). Turning on the triac at zero crossings eliminates EMI and reduces the turn-on stress in the triac. R₁ limits current into the triac driver (IC₂) to 1A. Each source can be either 24V ac (as shown) or 120V ac. (DI #2521)

Figure 2



TO VOTE FOR THIS DESIGN,
CIRCLE NO. 304

An ac switch (top trace) turns on and then briefly bounces off; the output of the associated debouncer circuit turns on cleanly (bottom trace).

Circuit detects reset source

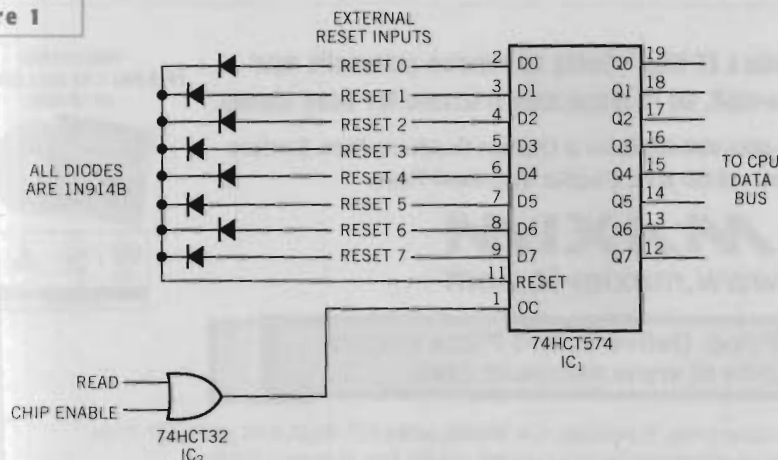
Shyam Tiwari, Sensors Technology Pvt Ltd, Gwalior, India

EMBEDDED SINGLE-BOARD computers do not rely only on power-on resets; they often use multiple sources to reset CPUs to prevent CPUs from being locked into endless loops. However, in

most cases, CPUs start from the starting points of memories to fetch their first code, because they have no way of knowing what generated the reset. The circuit in Figure 1 allows a CPU to know the

source of a reset. It stores in an 8-bit latch reset-control input data that the CPU can read. If an input frequently resets the CPU, the CPU can then report the error source to the user, using LEDs or other indicators. In the circuit of Figure 1, an experiment combined eight independent reset-signal sources into a wired-OR single output. The 74HCT574 stores the 8-bit reset data at the rising edge of the signal. The latch records no other signal if the signal appears after the rising edge of the first signal. If the CPU finds data that resembles a reset signal in the latch that the signal does not reset, then the reset pulse is too narrow to effect a reset. The CPU recognizes this signal after reading the information from the latch. The output of the latch is a tristate structure; the CPU reads the output using active chip-enable and read-input signals. (DI #2516)

Figure 1



This circuit detects the source of a CPU-reset signal.

TO VOTE FOR THIS DESIGN,
CIRCLE NO. 305